Office Action Summary	Application No.	Applicant(s)	
	10/629,810	TARIN, STEPHEN A.	
	Examiner	Art Unit	
	MAHESH H. DWIVEDI	2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 November 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *25-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *25-31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *28 July 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *3/5/07*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 9/5/2006 has been entered.

Response to Amendment

2. Receipt of Applicant's Amendment, filed on 11/09/2007, is acknowledged. The amendment includes the amending of claim 25, and the cancellation of claims 1-24.

Information Disclosure Statement

3. The information disclosure statements (IDS) submitted on 02/16/2007, 03/05/2007, and 03/05/2008 have been received, entered into the record, and considered. The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statements are being considered by the examiner.

The examiner notes that IDS submitted on 02/16/2007 incorrectly lists the author of reference C07. The author of that NPL reference is Nievergelt, not Nevergelt.

The examiner notes that IDS submitted on 03/05/2007 incorrectly lists the author of reference C07. The author of that NPL reference is Nievergelt, not Nevergelt.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 25-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Mehta** (U.S. Patent 5,999,933) and further in view of **Heising** (U.S. Patent 5,333,313).

6. Regarding claim 25, **Mehta** teaches a method comprising:

A) retrieving information regarding the number of occurrences of a given value from the database (Column 14, lines 4-17);

C) determining an instance element based on information regarding the number of occurrences of the given value (Column 14, lines 4-17);

D) determining a connectivity element based on the instance element (Column 20, lines 28-57); and

E) determining a record from the connectivity element (Column 21, lines 5-21).

The examiner notes that **Mehta** teaches **"retrieving information regarding the number of occurrences of a given value from the database"** as "An additional option in Table IX is fetch logic seven. Fetch logic seven is for determining the number of instances of data structures of a selected type from the number of instances of a reference data structure. In fetch logic seven, the number of instances is equal to the number of instances of another structure. Parameter Text1 is the template library name that contains the definition of the reference data structure. Parameter Text2 is the name of the reference data structure in the template library of parameter Text1. Parameter Int1 is the integer offset to parameter Text2 within the template library 152' designated by parameter Text1. The part in the template library 152' at offset Int1 describes a field in the reference data structure that contains the number of occurrences that this reference data structure has" (Column 14, lines 4-17). The examiner further notes that **Mehta** teaches **"determining an instance element based on information regarding the number of occurrences of the given value"** as "An additional option in Table IX is fetch logic seven. Fetch logic seven is for determining the number of instances of data structures of a selected type from the number of instances of a reference data structure. In fetch logic seven, the number of instances is equal to the number of instances of another structure. Parameter Text1 is the template library name that contains the definition of the reference data structure. Parameter Text2 is the name of the reference data structure in the template library of parameter Text1. Parameter Int1 is the integer offset to parameter Text2 within the template library 152' designated by parameter Text1. The part in the template library 152' at offset Int1 describes a field in the

reference data structure that contains the number of occurrences that this reference data structure has" (Column 14, lines 4-17). The examiner further notes that **Mehta** teaches **"determining a connectivity element based on the instance element"** as "In step 1646, the client 118 sends the pointer to the first instance to the server 114. The server 114 then fetches the first instance of the user selected type of data structure and sends it to the client 118. Then, the client 118 populates the first row in a logical table 153 with the first instance of the data structure. In step 1650, the client 118 executes the fetch logic function for obtaining the end logic and the links between instances of the selected type of data structure. The client 118 obtains iteratively the next instances of the data structure up to the last one. The client 118 does so by obtaining an instance of a data structure, storing it in the logical table 153, and obtaining from the last stored instance the pointer to the next instance of the data structure. Then the client 118 repeats this process with the next instance of the data structure. The client 118 uses the communication link 122 between the client 118 and the server 114 to request data from the server 114. The user can repeat the process of FIG. 16 for each type of data structure for which the user desires to create a logical table 153" (Column 20, lines 39-57). The examiner further notes that **Mehta** teaches **"determining a record from the connectivity element"** as "Following the creation of logical tables 153, the power of a standard database management system 146, such as MS Access, can be used to query these logical tables 153. As discussed above, the database management system 146 can be used to quickly and easily query logical tables 153 to determine for example, all PCB's with a priority of greater than one hundred. A standard database management system 146 has the ability to generate a query report. A report created by database management system 146 includes not only particular queries, but also their results. So, in this example, the report would include the query for all the PCB's of a priority greater than one hundred, and in addition it would include the actual PCB's of this priority. Many different queries (some of which were described further above) and their results can be stored in such a report. These reports are saved in the database 150, which is managed by the database management system 146" (Column 21, lines 5-21).

Mehta does not explicitly teach:

B) the database having been compressed by storing information regarding distinct values of an attribute and information regarding the number of occurrences of distinct values.

Heising, however, teaches **"the database having been compressed by storing information regarding distinct values of an attribute and information regarding the number of occurrences of distinct values"** as "The method divides the database information into a number of parts which are each conducive to a predetermined compression technique. A first part database is formed consisting of all the entry points in the dictionary wherein each entry point is associated with a unique word number. A second part database is formed consisting of a multiplicity of placeholders. A third part database is formed consisting of all the entry points of the dictionary in the exact order in which they appear in the dictionary. A fourth part database is formed consisting of the definitions and usage notes without reference to their text. A fifth part database allows retrieval of articles of interest without having to decompress the entire dictionary. Compression techniques using multigrams and minimum-redundancy codes are selectively applied to the different database parts" (Abstract), " The step of compressing the Lexicon is a technique which is described in Knuth. The general term for the structure used is a trie (sic). (See Knuth, The Art of Computer Programming, Vol. 3, Addison-Wesley Publishing Company; Reading, Mass.; 1973) " (Column 4, lines 5-9), and "The Forms database is extracted from the main database. Page breaks are kept in the same order as the main database. There tend to be many repeated sequences in this database. To take advantage of this, multigram codes are formed which substitute for the sequence. (A multigram is a code which represents a sequence of simpler codes; these simpler codes are referred to as unigrams. These codes will be discussed further below). A digram is a special case of multigram that represents exactly two unigrams. The method for selecting an optimal set of multigrams is explained in detail in a later section" (Column 4, lines 10-21).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching

Heising's would have allowed **Mehta's** to provide a method for making a database readably accessible after compression, as noted by **Heising** (Column 1, lines 33-40).

Regarding claim 26, **Mehta** further teaches a method comprising:
A) wherein the step of retrieving the information regarding the number of occurrences of the given value comprises analyzing the cardinality store (Column 14, lines 4-17).

The examiner notes that **Mehta** teaches **"wherein the step of retrieving the information regarding the number of occurrences of the given value comprises analyzing the cardinality store"** as "An additional option in Table IX is fetch logic seven. Fetch logic seven is for determining the number of instances of data structures of a selected type from the number of instances of a reference data structure. In fetch logic seven, the number of instances is equal to the number of instances of another structure. Parameter Text1 is the template library name that contains the definition of the reference data structure. Parameter Text2 is the name of the reference data structure in the template library of parameter Text1. Parameter Int1 is the integer offset to parameter Text2 within the template library 152' designated by parameter Text1. The part in the template library 152' at offset Int1 describes a field in the reference data structure that contains the number of occurrences that this reference data structure has" (Column 14, lines 4-17).

Regarding claim 27, **Mehta** further teaches a method comprising:
A) wherein the step of determining an instance element information regarding the number of occurrences of the given value comprises analyzing the instance store (Column 14, lines 4-17).

The examiner notes that **Mehta** teaches **"wherein the step of determining an instance element information regarding the number of occurrences of the given value comprises analyzing the instance store"** as "An additional option in Table IX is fetch logic seven. Fetch logic seven is for determining the number of instances of data structures of a selected type from the number of instances of a reference data structure. In fetch logic seven, the number of instances is equal to the number of instances of

another structure. Parameter Text1 is the template library name that contains the definition of the reference data structure. Parameter Text2 is the name of the reference data structure in the template library of parameter Text1. Parameter Int1 is the integer offset to parameter Text2 within the template library 152' designated by parameter Text1. The part in the template library 152' at offset Int1 describes a field in the reference data structure that contains the number of occurrences that this reference data structure has" (Column 14, lines 4-17).

Regarding claim 28, **Mehta** further teaches a method comprising:
A) wherein the step of determining the connectivity element comprises analyzing a connectivity store (Column 20, lines 28-57).

The examiner notes that **Mehta** teaches **"wherein the step of determining the connectivity element comprises analyzing a connectivity store"** as "In step 1646, the client 118 sends the pointer to the first instance to the server 114. The server 114 then fetches the first instance of the user selected type of data structure and sends it to the client 118. Then, the client 118 populates the first row in a logical table 153 with the first instance of the data structure. In step 1650, the client 118 executes the fetch logic function for obtaining the end logic and the links between instances of the selected type of data structure. The client 118 obtains iteratively the next instances of the data structure up to the last one. The client 118 does so by obtaining an instance of a data structure, storing it in the logical table 153, and obtaining from the last stored instance the pointer to the next instance of the data structure. Then the client 118 repeats this process with the next instance of the data structure. The client 118 uses the communication link 122 between the client 118 and the server 114 to request data from the server 114. The user can repeat the process of FIG. 16 for each type of data structure for which the user desires to create a logical table 153" (Column 20, lines 39-57).

Regarding claim 29, **Mehta** further teaches a method comprising:

A) wherein the step of determining a record comprises analyzing a value store (Column 21, lines 5-21).

The examiner notes that **Mehta** teaches **"wherein the step of determining a record comprises analyzing a value store"** as "Following the creation of logical tables 153, the power of a standard database management system 146, such as MS Access, can be used to query these logical tables 153. As discussed above, the database management system 146 can be used to quickly and easily query logical tables 153 to determine for example, all PCB's with a priority of greater than one hundred. A standard database management system 146 has the ability to generate a query report. A report created by database management system 146 includes not only particular queries, but also their results. So, in this example, the report would include the query for all the PCB's of a priority greater than one hundred, and in addition it would include the actual PCB's of this priority. Many different queries (some of which were described further above) and their results can be stored in such a report. These reports are saved in the database 150, which is managed by the database management system 146" (Column 21, lines 5-21).

7. Claims 30-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Mehta** (U.S. Patent 5,999,933) and **Heising** (U.S. Patent 5,333,313) as applied to claims 25-29 and in view of **White et al.** (U.S. Patent 5,918,225).

8. Regarding claim 30, **Mehta** and **Heising** do not explicitly teach a method comprising:

A) wherein the retrieving of the record is causes by an Structured Query Language (SQL) query.

White, however, teaches **"wherein the retrieving of the record is causes by an Structured Query Language (SQL) query"** as "Clients store data in and retrieve data from one or more database tables resident on the Server by submitting SQL commands" (Abstract)

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **White's** would have allowed **Mehta's** and **Heising's** to provide a method for improved

performance in data retrieval for relational databases via SQL, as noted by **White** (Column 3, lines 1-7).

Regarding claim 31, **Mehta** and **Heising** do not explicitly teach a method comprising:

A) wherein the SQL query is a SELECT query.

White, however, teaches **"wherein the SQL query is a SELECT query"** Here, "projecting" data refers to the project part of an SQL SELECT statement" (Column 48, lines 42-43)

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **White's** would have allowed **Mehta's** and **Heising's** to provide a method for improved performance in data retrieval for relational databases via SQL, as noted by **White** (Column 3, lines 1-7).

Response to Arguments

9. Applicant's arguments with respect to claim 25 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

10. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Patent 5,546,575 issued to **Potter et al.** on 13 August 1996. The subject matter disclosed therein is pertinent to that of claims 25-31 (e.g., methods to query compressed databases).

U.S. Patent 5,592,667 issued to **Bugajski** on 07 January 1997. The subject matter disclosed therein is pertinent to that of claims 25-31 (e.g., methods to query compressed databases).

U.S. Patent 5,946,692 issued to **Faloutsos et al.** on 31 August 1999. The subject matter disclosed therein is pertinent to that of claims 25-31 (e.g., methods to query compressed databases).

U.S. Patent 6,006,232 issued to **Lyons** on 21 December 1999. The subject matter disclosed therein is pertinent to that of claims 25-31 (e.g., methods to query compressed databases).

Contact Information

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mahesh Dwivedi whose telephone number is (571) 272-2731. The examiner can normally be reached on Monday to Friday 8:20 am – 4:40 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached (571) 272-3642. The fax number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mahesh Dwivedi
Patent Examiner
Art Unit 2168

February 13, 2008